

CLAIMS

What is claimed is:

1. A method for retrieving data comprising:

locking a linked list;

5

retrieving data from an element in the linked list and also

advancing to a subsequent element while a breakpoint is not encountered;

marking the subsequent element in the linked-list as in-use when a breakpoint is

encountered;

creating a recommencement reference to the subsequent element; and

10

unlocking the linked list.

2. The method of Claim 1 further comprising:

locking the linked list;

determining a subsequent element in the linked list according to the recommencement

reference; and

15

retrieving data from the determined subsequent element.

3. The method of Claim 1 wherein creating a recommencement reference to the subsequent element comprises:

retrieving a pointer to the subsequent element;

determining a process identifier for a current process; and

20

associating the pointer with the process identifier.

4. The method of Claim 1 wherein marking the subsequent element in the linked-list as in-use comprises maintaining a count of the quantity of processes that require additional access to the element.

25

5. A method for deleting an element from a linked list comprising:

determining if the element to be deleted is in-use;
updating a recommencement reference to the element to refer to a data element that is
subsequent to the data element to be deleted when the element in is in-use; and
5 deleting the element.

6. The method of Claim 5 wherein updating a recommencement reference to the element
comprises:

discovering a pointer associated with a process identifier;
disassociating the process identifier from the pointer;
10 determining a pointer to a subsequent element; and
associating the process identifier with the newly determined pointer.

7. An apparatus for storing and retrieving data comprising:

processor capable of executing an instruction sequence;
memory for storing an instruction sequence;
15 input unit for receiving data;
first output unit for providing data according to a received data request;
one or more ancillary output units for providing data according to a received data
request;

instruction sequences stored in the memory including:

20 data storage module that, when executed by the processor, minimally causes
the processor to:

receive data from the input unit;
allocate a data element to accommodate the data;
create a reference to the data element;

25 store the reference in at least one of a header pointer and a forward
pointer included in a preceding data element; and
store the data in the data element;

data service module that, when executed by the processor, minimally causes
the processor to:

recognize a data request from the first output unit to the exclusion of
all other data requests;
provide data to the first output unit from a data element according to a
data element reference and also
5 advance the data element reference to a subsequent data element while
a breakpoint is not encountered;
mark a subsequent data element as in-use when a breakpoint is
encountered;
create a recommencement reference to a subsequent data element; and
10 enable recognition of other data requests.

8. The apparatus of Claim 7 wherein the data service module, when executed by the
processor, further minimally causes the processor to:

recognize a data request from the first output unit to the exclusion of all other data
requests; and
15 provide data to the first output unit from a data element according to the
recommencement reference.

9. The apparatus of Claim 7 wherein the data service module causes the processor to create
a recommencement reference by minimally causing the processor to:

retrieve a pointer to a data element subsequent to a current data element;
20 determine an identifier associated with the data request received from the first output
unit; and
store the retrieved pointer and the determined identifier in an associative manner.

10. The apparatus of Claim 7 wherein the data service module causes the processor to mark a
subsequent data element as in-use by minimally causing the processor to increment a use
25 counter included in a subsequent data element.

11. The apparatus of Claim 7 wherein the data service module further minimally causes the
processor to receive a delete data request from an output unit by minimally causing the
processor to:

determine if a data element to be deleted is in-use;
update a recommencement reference to refer to a data element that is subsequent to
the data element to be deleted; and
delete the data element according to the received delete data request.

- 5 12. The apparatus of Claim 11 wherein the data service module causes the processor to
update a recommencement reference by minimally causing the processor to:
discover a pointer according to a data request identifier; and
replace the pointer with a pointer to a data element that is subsequent to the data
element to be deleted.

- 10 13. A computer readable medium having imparted thereon one or more instruction sequences
for storing and retrieving data comprising:

data storage module that, when executed by a processor, minimally causes the
processor to:

- 15 receive data from an input unit;
allocate a data element to accommodate the data;
create a reference to the data element;
store the reference in at least one of a header pointer and a forward
pointer included in a preceding data element; and
store the data in the data element;

- 20 data service module that, when executed by a processor, minimally causes the
processor to:

- 25 recognize a data request from a first output unit to the exclusion of all
other data requests;
provide data to a first output unit from a data element according to a
data element reference and also
advance the data element reference to a subsequent data element while
a breakpoint is not encountered;
mark a subsequent data element as in-use when a breakpoint is
encountered;

create a recommencement reference to a subsequent data element; and
enable recognition of other data requests.

14. The computer readable medium of Claim 13 wherein the data service module, when
executed by a processor, further minimally causes the processor to:

5 recognize a data request from a first output unit to the exclusion of all other data
 requests; and
 provide data to a first output unit from a data element according to the
 recommencement reference.

15. The computer readable medium of Claim 13 wherein the data service module causes a
10 processor to create a recommencement reference by minimally causing the processor to:
 retrieve a pointer to a data element subsequent to a current data element;
 determine an identifier associated with a data request received from a first output unit;
 and
 store the retrieved pointer and the determined identifier in an associative manner.

15 16. The computer readable medium of Claim 13 wherein the data service module causes a
 processor to mark a subsequent data element as in-use by minimally causing the
 processor to increment a use counter included in a subsequent data element.

17. The computer readable medium of Claim 13 wherein the data service module further
minimally causes the processor to receive a delete data request from an output unit by
20 minimally causing the processor to:
 determine if a data element to be deleted is in-use;
 update a recommencement reference to refer to a data element that is subsequent to
 the data element to be deleted; and
 delete the data element according to the received delete data request.

25 18. The computer readable medium of Claim 17 wherein the data service module causes the
 processor to update a recommencement reference by minimally causing the processor to:
 discover a pointer according to a data request identifier; and

replace the pointer with a pointer to a data element that is subsequent to the data element to be deleted.

19. An apparatus for storing and retrieving data comprising:

means for locking a linked list;

5 means for retrieving data from an element in the linked list and also
advancing to a subsequent element while a breakpoint is not encountered;

means for marking the subsequent element in the linked-list as in-use when a
breakpoint is encountered;

means for creating a recommencement reference to the subsequent element; and

10 means for unlocking the linked list.

20. The apparatus of Claim 19 further comprising:

means for locking the linked list;

means for determining a subsequent element in the linked list according to the
recommencement reference; and

15 means for retrieving data from the determined subsequent element.

21. The apparatus of Claim 19 further comprising a means for deleting an element in the
linked-list.

22. The apparatus of Claim 21 wherein the means for deleting an element comprises:

means for determining if the element to be deleted is in-use;

20 means for updating a reference to the element to refer to a subsequent element in the
linked list when the element is in-use; and

means for deleting the element.